Chapter 1
Multi-objective Evolutionary Algorithms in Real-World Applications: Some Recent Results and Current Challenges

Carlos A. Coello Coello

Abstract This chapter provides a short overview of the most significant research work that has been conducted regarding the solution of computationally expensive multi-objective optimization problems. The approaches that are briefly discussed include problem approximation, function approximation (i.e., surrogates) and evolutionary approximation (i.e., clustering and fitness inheritance). Additionally, the use of alternative approaches such as cultural algorithms, small population sizes and hybrids that use a few solutions (generated with optimizers that sacrifice diversity for the sake of a faster convergence) to reconstruct the Pareto front with powerful local search engines are also briefly discussed. In the final part of the chapter, some topics that (from the author's perspective) deserve more research, are provided.

Keywords Evolutionary algorithms · Multi-objective optimization · Metaheuristics

1.1 Introduction

In real-world applications, most problems have several (often conflicting) objectives that we aim to optimize at the same time. Such problems are called "multi-objective" and their solution gives rise to a set of solutions that represent the best possible trade-offs among all the objectives (i.e., the so-called *Pareto optimal set*). The image of the Pareto optimal set (i.e., the objective function values corresponding to this set) forms to so-called *Pareto front* of the multi-objective optimization problem being solved.

Starting in the mid-1980s, Evolutionary Algorithms (EAs) have become a popular search engine to solve multi-objective optimization problems, mainly because of their ease of use, and wide applicability (i.e., they require little domain-specific information to operate) [11, 15].

The author acknowledges the financial support obtained through a "Cátedra Marcos Moshinsky".

C.A. Coello Coello (✉)
CINVESTAV-IPN (Evolutionary Computation Group), 07360 Mexico, D.F., Mexico
e-mail: ccoello@cs.cinvestav.mx

D. Greiner et al. (eds.), *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, Computational Methods in Applied Sciences 36, DOI 10.1007/978-3-319-11541-2_1

Modern multi-objective evolutionary algorithms (MOEAs) consist of two main components:

1. A selection mechanism that is normally (but not necessarily) based on Pareto optimality. Performance indicators can also be used for selecting solutions in a population and that has been, indeed, a relatively popular research trend in recent years [3].
2. A density estimator, which is responsible for producing different elements of the Pareto optimal set in a single run of a MOEA. Different options are available for this mechanism, such as: fitness sharing [16], entropy [82], clustering [79], adaptive grids [31] and crowding [17], among others.

Additionally, all modern MOEAs are *elitist*, which means that they retain the nondominated solutions generated at each iteration, so that at the end of a run, the user can have the globablly nondominated solutions that had been produced. Elitism is normally implemented through the use of an external archive, but the use of the main population for this purpose is also possible.

In spite of their popularity, one of the main limitations of MOEAs, when used for solving real-world problems, is their high computational cost, which is associated to the relatively high number of objective function evaluations that most current MOEAs require [62]. Although there are several remarkable efforts in this regard, several challenges still lie ahead, and the purpose of this chapter is precisely to review some of the most representative research that has been conducted in this area.

The remainder of this chapter is organized as follows. In Sect. 1.2, we present basic concepts related to multi-objective optimization. Then, in Sect. 1.3, we discuss the main schemes that have been proposed for dealing with expensive multi-objective optimization problems. In Sect. 1.4, we explore other ideas that havel also been used for dealing with real-world applications having objective functions that are computationally expensive. Section 1.5, provides some potential paths for future research in this area. Finally, the conclusions of this chapter are presented in Sect. 1.6.

1.2 Basic Concepts

We are interested in solving problems of the type[1]:

$$\text{minimize}\, \mathbf{f}(\mathbf{x}) := [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})] \tag{1.1}$$

subject to:

$$g_i(\mathbf{x}) \leq 0 \quad i = 1, 2, \ldots, m \tag{1.2}$$

$$h_i(\mathbf{x}) = 0 \quad i = 1, 2, \ldots, p \tag{1.3}$$

[1] Without loss of generality, we will assume only minimization problems.

where $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ is the vector of decision variables, $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, k$ are the objective functions and $g_i, h_j : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, m$, $j = 1, \ldots, p$ are the constraint functions of the problem.

To describe the concept of optimality in which we are interested, we will introduce next a few definitions.

Definition 1 Given two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^k$, we say that $\mathbf{x} \leq \mathbf{y}$ if $x_i \leq y_i$ for $i = 1, \ldots, k$, and that **x dominates y** (denoted by $\mathbf{x} \prec \mathbf{y}$) if $\mathbf{x} \leq \mathbf{y}$ and $\mathbf{x} \neq \mathbf{y}$.

Definition 2 We say that a vector of decision variables $\mathbf{x} \in \mathscr{X} \subset \mathbb{R}^n$ is **nondominated** with respect to $\mathscr{X}$, if there does not exist another $\mathbf{x}' \in \mathscr{X}$ such that $\mathbf{f}(\mathbf{x}') \prec \mathbf{f}(\mathbf{x})$.

Definition 3 We say that a vector of decision variables $\mathbf{x}^* \in \mathscr{F} \subset \mathbb{R}^n$ ($\mathscr{F}$ is the feasible region) is **Pareto-optimal** if it is nondominated with respect to $\mathscr{F}$.

Definition 4 The **Pareto Optimal Set** $\mathscr{P}^*$ is defined by:

$$\mathscr{P}^* = \{\mathbf{x} \in \mathscr{F} | \mathbf{x} \text{ is Pareto-optimal}\}$$

Definition 5 The **Pareto Front** $\mathscr{PF}^*$ is defined by:

$$\mathscr{PF}^* = \{\mathbf{f}(\mathbf{x}) \in \mathbb{R}^k | \mathbf{x} \in \mathscr{P}^*\}$$

Therefore, we wish to determine the Pareto optimal set from the set $\mathscr{F}$ of all the decision variable vectors that satisfy (1.2) and (1.3). In practice, however, not all the Pareto optimal set is normally desirable or even achievable.

1.3 Dealing with Expensive Problems

In general, MOEAs can be unaffordable for an application when:

- The evaluation of the fitness functions is computationally expensive (e.g., it takes several hours).
- The total number of fitness function evaluations that can be performed is limited (e.g., we only have a certain computational budget available).

According to [29], there are three main schemes that can be used to deal with expensive problems:

Problem approximation: In this case, the idea is to replace the original (expensive) statement of the problem by another one which is easier (and less expensive) to solve.

Functional approximation: In this case, instead of using the original objective function(s) (which is/are expensive to evaluate), an alternative expression(s) is adopted. The new expression(s) is built based on the previous data obtained from evaluating the real objective function(s). The models that are obtained from the data that is currently available are called *meta-models* or *surrogates*.

Evolutionary approximation: This is an approach that is specific to EAs, and that aims to save fitness function evaluations by estimating the fitness of an individual using information from other (similar) individuals. The two main approaches in this class are: fitness inheritance and clustering.

Next, we will provide a short discussion of each of these schemes, as well as some real-world problems in which they have been adopted.

1.3.1 Use of Problem Approximation

As indicated before, in this case, the idea is to replace the original problem by another one which is easier to solve. This sort of approach has been relatively popular in aeronautical/aerospace engineering, in which complex Computational Fluid Dynamics (CFD), Computational Aero-Acoustics (CAA) and Computational Structural Mechanics (CSM) are adopted. When using such tools, it is possible to approximate the original problem by using different resolutions in the flow or structural simulation, adopting either coarse or fine grids. For CFD simulations is also possible to rely on Euler flows instead of (the more expensive) Navier-Stokes flow simulations.

An example of this sort of approach is the work of Lee et al. [41, 42]. In this case, the authors applied the HAPMOEA (Hierarchical Asynchronous Parallel Multi-Objective Evolutionary Algorithm) [24] to the robust design optimization of an ONERA M6 wing shape. The authors considered uncertainties in the design environment, related to the flow Mach number, and the Taguchi method was used to transform the problem into one with two objectives to be minimized: (1) the mean value of an objective function with respect to variability of the operating conditions, and (2) the variance of the objective function of each solution candidate, with respect to its mean value. HAPMOEA uses an evolution strategy as its search engine, incorporating the concept of Covariance Matrix Adaptation (CMA). It also incorporates a distance-dependent mutation operator, and a hierarchical set of CFD models (varying the grid resolution of the solver). Small populations are evolved using fine mesh CFD solutions in order to exploit the search space, while large populations are evolved with coarse mesh CFD solutions for exploring the search space. Good solutions from the coarse mesh populations (in which evaluations have a low computational cost) are transferred to the fine mesh populations (in which evaluations are computationally expensive).

For more information on this topic, the interested reader must refer to: [6, 62, 68]

1.3.2 Use of Functional Approximation

The use of meta-models or surrogate models has been very popular in engineering. In order to build a meta-model, a set of data points that lie on the local neighborhood of the design is required. The accuracy of the meta-model relies on the number of samples provided (from the real objective function evaluations), as well as on the accuracy of the model that is used to approximate the objective functions. Such an approximate model must also have a low computational cost, since it will be evaluated many times during the search.

There are several techniques available for constructing surrogate models, from which the main ones are [62]: response surface methods, Gaussian processes (or Kriging), radial basis functions, artificial neural networks and support vector machines.

An example of this sort of approach is the work of Voutchkov et al. [81], in which the Nondominated Sorting Genetic Algorithm-II (NSGA-II) [17] was used to perform a robust structural design of a simplified jet engine model. The aim was to find the best jet engine structural configuration that minimized: the variation of reacting forces under a range of external loads, the mass for the engine and the engine's fuel consumption. The evaluation of the structural response was done in parallel by means of finite element simulations. The authors adopted a kriging based response surface method in order to reduce the computational time required to solve this problem. Four objectives were minimized: (1) standard deviation of the internal reaction forces, (2) mean value of the internal reaction forces, (3) engine's mass, and (4) mean value of the specific fuel consumption. The first two objectives were computed over 200 external load variations. Due to the many combinations of loads and finite element thicknesses, the multi-objective optimization problem would have taken on the order of one year of computational time on a single 1 GHZ CPU, if no effort had been made to perform a more efficient search. When using the surrogate model that they report, combined with parallel processing, the total optimization time was reduced to about 26 h, in a cluster with 30 cores.

For more information on this topic, the interested reader must refer to: [32, 43, 45, 51, 53].

1.3.3 Use of Evolutionary Approximation

In this case, two main approaches are considered: clustering and fitness inheritance. Next, we will briefly discuss each of them.

Clustering is a term used to refer to the unsupervised classification of paterns into groups (which are called *clusters*). The idea is to partition data into different groups either in a hard way (i.e., into well-defined groups) or in a fuzzy way (i.e., using a certain degree of membership to each of the groups) [27].

Although clustering is normally not used as a specific technique to reduce objective function evaluations, this sort of technique is normally adopted in combination with surrogates in order to reduce the size of the training data set. This is an important task, since the use of very large training data sets makes prohibitive the cost of a surrogate method. Clustering is normally adopted in this context to split the data set into several small groups, and then an independent local model is built from each of them.

An example of the use of clustering is the work of Langer et al. [38], in which an integrated approach that adopts computer aided design modeling is combined with a MOEA for solving structural shape and topology optimization problems. The authors were interested in optimizing an instrument panel of a satellite, considering two objectives: (1) minimize the instrument panel mass, and (2) maximize the first eigenfrecuency. The authors solved the optimization problem for three shape and topology optimization cases: (a) a panel without instruments, (b) a panel with instruments at fixed positions, and (c) a panel with instrumental placing. They adopted polynomial based response surface methods in order to reduce the computational cost, and multiple local approximation models were constructed using a clustering technique. The use of parallel techniques was also required in this case (a cluster with 32 processors was adopted by the authors).

Fitness inheritance was originally introduced by Smith et al. [71], with the motivation of reducing the total number of fitness function evaluations performed by an evolutionary algorithm. The idea is that, when assigning fitness to an individual, some times we evaluate the objective function as usual, but the rest of the time, we assign a fitness value equal to the average of the fitness values of its parents. This saves one fitness function evaluation, and is based on the assumption of similarity of an offspring to its parents.

Evidently, fitness inheritance cannot be applied all the time, since it is required to have information from true fitness function evaluations in order to guide the search in a proper way. The percentage of time in which fitness inheritance is applied is called *inheritance proportion*. Clearly, this proportion should be less than one in order to avoid premature convergence [4].

A theoretical model of fitness inheritance was presented by Sastry et al. [69]. Such model was used to obtain the convergence time, the optimal population size and the optimal inheritance proportion (the authors found that values between 0.54 and 0.558 worked best for the inheritance proportion in problems of moderate and large size).

The work of Sastry et al. [69] was extended to the multi-objective case by Chen et al. [4]. In this case, the authors used fitness sharing to maintain diversity in the population with the aim of covering a larger extension of the Pareto front. The problem they solved was a bi-objective extension of the OneMax problem originally solved by Sastry et al. [69] in their study. The authors also presented a generalization (for the multi-objective case) of the theoretical work reported by Sastry et al. [69] regarding convergence time, optimal population sizing and optimal inheritance proportion. The experiments reported by the authors showed that savings of up to 40 % of the total number of evaluations could be achieved when using fitness inheritnance alone. When combining fitness inheritance with fitness sharing, savings of up to 25 % were obtained.

Reyes-Sierra and Coello Coello proposed the use of dynamic rules to assign the inheritance proportion in a multi-objective particle swarm optimizer [55]. Such rules produced savings that were from 19 up to 78 % of the total number of evaluations. However, as expected, the greater the savings in the number of evaluations, the greater was the degradation in the quality of the results. Nevertheless, the authors showed it was possible to obtain savings of up to 49 % without having a significant loss in the quality of the results. The authors adopted the Zitzler-Deb-Thiele (ZDT) test problems in their experiments [89].

It is worth mentioning that some researchers have considered fitness inheritance to be an inappropriate mechanism in complex or real-world problems (see for example [20], in which the authors concluded that fitness inheritance was not useful when the shape of the Pareto front is nonconvex or discontinuous). Such conclusions are valid for the proposal reported in [20]. However, in [56] it is shown that these limitations of fitness inheritance can be overcome, so that this approach can be applied to Pareto fronts having any kind of shape.

For more information on this topic, the interested reader must refer to: [19, 23, 37, 52, 56].

1.4 Other Approaches

There are some other ideas that can be used to tackle problems with computationally expensive objective functions, and which do not fall into any of the categories analyzed in the previous section. Here, we will focus on three types of approaches:

1. Cultural algorithms
2. Use of very small population sizes
3. Use of efficient search techniques

Next, we will briefly discuss each of these three types of approaches.

1.4.1 Cultural Algorithms

Cultural algorithms were originally proposed by Robert Reynolds in the mid-1990s [57, 60]. The core idea behind cultural algorithms is to incorporate domain knowledge extracted during the search to an evolutionary algorithm. Cultural algorithms use, in addition to the population space commonly adopted in evolutionary algorithms, a belief space, which encodes the knowledge obtained from the search points that have been evaluated so far. The belief space is used to influence the evolutionary operators, with the aim of guiding the search in a very efficient way.

At each generation, a cultural algorithm selects some individuals from the population, in order to extract information from them. Such information will then be used to speed up the search. Evidently, the belief space requires some sort of scheme to represent the knowledge extracted during the evolutionary process and this representation is normally specific for each particular problem (or class of problems). It is also necessary to design mechanisms that allow to use this extracted knowledge to influence the way in which the evolutionary operators explore and exploit the search space.

Although cultural algorithms have been adopted for single-objective optimization by several authors (see for example [7, 28, 35, 58, 59]), their use in multi-objective optimization has been very limited until now.

The first proposal to design a cultural algorithm for solving multi-objective optimization problems is the framework described in [12], which uses Pareto ranking, and an approximation of the dimensions of the Pareto front in the belief space. In this proposal, the belief space works as a guide for the individuals to reach regions where nondominated solutions have been found. The belief space includes also a mechanism to obtain a good distribution of the resulting points along the Pareto front (i.e., a density estimator).

The earliest attempt to solve multi-objective optimization problems using cultural algorithms was based on the use of the ε-constraint method [36], since this sort of approach uses a single-objective optimizer rather than a MOEA (the cultural algorithm with differential evolution proposed in [35] was adopted for this sake). This approach turned out to be computational expensive, due to the high number of objective function evaluations required to generate a good approximation of the Pareto front. However, the authors showed that if the aim was to solve very difficult multi-objective optimization problems, then this additional computational cost was worth it. This was illustrated by solving several problems from the Deb-Thiele-Laumanns-Zitzler (DTLZ) [18] and the Walking-Fish-Group (WFG) [25, 26] test suites.

More recently, Best and his collaborators [1, 2] proposed a more general framework for using cultural algorithms with any sort of MOEA. This approach is interesting and incorporates several sources of knowledge, but it did not show a significant reduction of objective function evaluations, which is one of the main motivations for using cultural algorithms. Additionally, the results presented by the authors are not competitive with respect to those obtained by traditional MOEAs using the same number of objective function evaluations, which suggests that it is still required to conduct more research in this area. In fact, the incorporation of knowledge into MOEAs (using any sort of scheme), with the aim of making them more efficient is indeed a very promising research area [37].

1.4.2 Use of Very Small Population Sizes

The use of small population sizes is unusual in the evolutionary algorithms literature in general, mainly because of the evident loss of diversity that is associated to small population sizes, and which normally leads to premature convergence. However, in the genetic algorithms literature, it is known that the use of very small population sizes is possible, if an appropriate reinitialization process is adopted (such approaches are called micro-genetic algorithms (micro-GAs) [13, 14, 34] and they use populations with no more than five individuals).

Krishnakumar [34] proposed the first implementation of a micro-GA. The first micro-GA for multi-objective optimization was introduced in [13, 14]. This approach uses a population size of four individuals, and three forms of elitism: (1) an external archive that adopts the adaptive grid from the Pareto Archived Evolution Strategy (PAES) [33], (2) a population memory, in which randomly generated individuals are replaced by evolved individuals, and (3) a mechanism that retains the two best solutions generated by each run of the micro-GA. The main advantage of this approach is its efficiency (its authors showed that their approach was up to an order of magnitude faster than the NSGA-II [17]). This is the reason why this approach has been used in computationally expensive real-world applications (see for example [8, 9]).

In a further paper, Coello Coello and Pulido introduced the micro-GA^2 [78], which is a fully self-adaptive MOEA that adopts a parallel strategy to adapt the crossover operator and the type of encoding (binary or real numbers) to be used. This approach can even stop automatically (it uses a mechanism based on a performance indicator to decide when to stop the search).

Over the years, other authors have adopted micro-genetic algorithms for solving a variety of problems (see for example [5, 30, 46, 47, 61, 72, 76, 77]). Additionally, the use of very small population sizes has also been attempted with other bio-inspired metaheuristics, such as particle swarm optimization (see [22]).

1.4.3 Use of Efficient Search Techniques

During the last few years, some researchers have proposed schemes that allow a more efficient exploration of the search space through the use of aggressive search engines that produce a few points from the Pareto front and then adopt a local search engine to reconstruct the rest of the front. One example of this sort of hybrid MOEA is DEMORS (differential evolution (DE) for multi-objective optimization with local search based on rough set theory) [64]. This approach operates in two phases. In the first one, a DE-based MOEA produces a rough approximation of the Pareto front using a relatively low number of objective function evaluations (65 % of the total number of objective function evaluations adopted by DEMORS are spent in the first phase). In the second phase, the remainder 35 % of objective function evaluations still available, are spent on the use of a local search procedure based on rough set theory [49, 50], whose task is to reconstruct the missing parts of the Pareto front. DEMORS was validated using several standard test problems taken from the specialized literature, as well as in a real-world problem having 8 objective functions and 160 decision variables in which it was able to outperform NSGA-II.

The same authors experimented with other (similar) hybrids in which DE was replaced by particle swarm optimization [66, 67] or rough sets were replaced by scatter search [65]. All these approaches were found to be very efficient multi-objective optimizers, and seem particularly suitable for real-world applications in which the use of surrogates is not appropriate.

In further related work [63], the same authors compared different surrogate methods (namely, artificial neural networks, radial basis functions and support vector machines) coupled to a MOEA and combined the best performer of them (support vector machines) with rough sets. This sort of scheme was proposed as an alternative for dealing with multi-objective problems that are very expensive (computationally speaking).

1.5 Future Research Paths

There are a number of possible research paths in this area that are worth exploring:

- **Parallel Approaches**: Although parallel MOEAs have been used for several years [11, 48], most of the papers published in that area focus on discussing applications and normally, such papers put little emphasis on the development of innovative algorithmic designs. Nowadays, the use of grid computing and Graphics Processing Units (GPUs) opens new and promising venues for future research in this area (see for example [21, 70, 73–75, 80, 88]), particularly regarding the solution of real-world problems having computationally expensive objective functions. The incorporation of surrogate models into parallel MOEAs is another interesting topic that deserves more research and that has been only scarcely explored in the specialized literature until now (see for example [54]).
- **Hybridization**: Coupling gradient-based or direct search methods to MOEAs is another alternative way for dealing with computationally expensive problems. In recent years, several promising hybrids of this sort have been proposed (see for example [39, 40, 44, 83–85]). These approaches can also be combined with surrogates for further efficiency (see for example [86, 87]). However, the use of such hybrid approaches in real-world applications is still rare (see for example [8]). Nevertheless, this situation is expected to change as more research results in this area become available.
- **Sampling techniques**: Surrogate methods heavily rely on the sample and updating technique adopted. In many real-world applications that use surroages, latin hypercubes have been adopted for the initial sampling, with the aim of covering as much as possible of the design (i.e., decision variable) space. At later stages of the search, it may be more relevant to explore the neighborhood of a good solution (see for example [8]). However, sampling is also relevant in other approaches, such as when using small population sizes or when hybridizing a MOEA with a local search engine. Nevertheless, the impact of the sampling technique in the performance of such approaches has not been properly addressed so far, to the author's best knowledge.

1.6 Conclusions

This chapter has provided a quick overview of the most relevant research tools that are currently available to tackle computationally expensive problems using multi-objective evolutionary algorithms. Breadth has been emphasized over depth in the discussions provided herein. However, several additional references have been provided for those interested in getting an in-depth knowledge about any of the topics that have been addressed in this chapter.

One aspect that is worth mentioning is that the presence of computationally expensive objective functions is clearly not the only relevant aspect when solving real-world problems. Other issues such as scalability (in decision variable space or in objective function space, or in both), uncertainty and incorporation of user's preferences, just to name a few, have not been addressed here, mainly because of obvious space limitations. Readers interested in information about these and other relevant topics are invited to visit the EMOO repository [10], which is available at: http://delta.cs.cinvestav.mx/~ccoello/EMOO/EMOObib.html.

References

1. Best C (2009) Multi-objective cultural algorithms. Master's thesis. Wayne State University, Detroit, Michigan, USA
2. Best C, Che X, Reynolds RG, Liu D (2010) Multi-objective cultural algorithms. In: 2010 IEEE congress on evolutionary computation (CEC'2010), Barcelona, Spain, 18–23 July 2010, IEEE Press, pp 3330–3338
3. Nicola B, Naujoks B, Emmerich M (2007) SMS-EMOA: multiobjective selection based on dominated hypervolume. Eur J Oper Res 181(3):1653–1669
4. Chen J-H, Goldberg DE, Ho S-Y, Sastry K (2002) Fitness inheritance in multi-objective optimization. In: Langdon WB, Cantú-Paz E, Mathias K, Roy R, Davis D, Poli R, Balakrishnan K, Honavar V, Rudolph G, Wegener J, Bull L, Potter MA, Schultz AC, Miller JF, Burke E, Jonoska N (eds) Proceedings of the genetic and evolutionary computation conference (GECCO'2002), Morgan Kaufmann Publishers, San Francisco, California, July 2002, pp 319–326
5. Chen Y, Ma Y, Lu Z, Qiu L, He J (2011) Terahertz spectroscopic uncertainty analysis for explosive mixture components determination using multi-objective micro-genetic algorithm. Adv Eng Softw 42(9):649–659
6. Chiba K, Obayashi S, Nakahashi K, Morino H (2005) High-fidelity multidisciplinary design optimization of wing shape for regional jet aircraft. In: Coello Coello CA, Aguirre AH, Zitzler E (eds) Evolutionary multi-criterion optimization. Third international conference, EMO 2005, Guanajuato, México, March 2005, Lecture notes in computer science. Springer, Heidelberg, pp 621–635
7. Chung C-J, Reynolds RG (1998) CAEP: an evolution-based tool for real-valued function optimization using cultural algorithms. J Artif Intell Tools 7(3):239–292
8. Chung HS (2004) Multidisciplinary design optimization of supersonic business jets using approximation model-based genetic algorithms. PhD thesis, Department of Aeronautics and Astronautics, Stanford University, California, USA, March 2004
9. Chung H-S, Alonso JJ (2004) Multiobjective optimization using approximation model-based genetic algorithms. In: Proceedings of the 10th AIAA/ISSMO symposium on multidisciplinary analysis and optimization, Albany, New York, USA, September 2004. Paper AIAA-2004-4325
10. Coello Coello CA (2006) The EMOO repository: a resource for doing research in evolutionary multiobjective optimization. IEEE Comput Intell Mag 1(1):37–45
11. Coello Coello CA, Lamont GB, Van Veldhuizen DA (2007) Evolutionary algorithms for solving multi-objective problems, 2nd edn. Springer, New York. ISBN 978-0-387-33254-3
12. Coello Coello CA, Becerra RL (2003) Evolutionary multiobjective optimization using a cultural algorithm. In: 2003 IEEE swarm intelligence symposium proceedings, Indianapolis, Indiana, USA, April 2003, IEEE Service Center, pp 6–13
13. Coello Coello CA, Pulido GT (2001) A micro-genetic algorithm for multiobjective optimization. In: Zitzler E, Deb K, Thiele L, Coello Coello CA, Corne D (eds) First international conference on evolutionary multi-criterion optimization. Lecture notes in computer science. Springer, Heidulberg, pp 126–140
14. Coello Coello CA, Pulido GT (2001) Multiobjective optimization using a micro-genetic algorithm. In: Spector L, Goodman ED, Wu A, Langdon WB, Voigt H-M, Gen M, Sen S, Dorigo M,

Pezeshk S, Garzon MH, Burke E (eds) Proceedings of the genetic and evolutionary computation conference (GECCO'2001). Morgan Kaufmann, San Francisco, pp 274–282
15. Deb K (2001) Multi-objective optimization using evolutionary algorithms. Wiley, Chichester ISBN 0-471-87339-X
16. Deb K, Goldberg DE (1989) An investigation of niche and species formation in genetic function optimization. In: Schaffer JD (ed) Proceedings of the third international conference on genetic algorithms, San Mateo, California, June 1989, George Mason University. Morgan Kaufmann, Burlington, pp 42–50
17. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197
18. Deb K, Thiele L, Laumanns M, Zitzler E (2005) Scalable test problems for evolutionary multi-objective optimization. In: Abraham A, Jain L, Goldberg R (eds) Evolutionary multiobjective optimization. Theoretical advances and applications. Springer, Heidelberg, pp 105–145
19. Ducheyne EI, De Baets B, De Wulf RR (2008) Fitness inheritance in multiple objective evolutionary algorithms: a test bench and real-world evaluation. Appl Soft Comput 8(1):337–349
20. Ducheyne EI, Baets BD, De Wulf R (2003) Is fitness inheritance useful for real-world applications? In: Fonseca CM, Fleming PJ, Zitzler E, Deb K, Thiele L (eds) Evolutionary multi-criterion optimization. Second international conference, EMO 2003, Faro, Portugal, April 2003. Lecture notes in computer science. Springer, Heidelberg, pp 31–42
21. Grzegorz E, Wojciech K, Brdys MA (2008) Grid implementation of a parallel multiobjective genetic algorithm for optimized allocation of chlorination stations in drinking water distribution systems: Chojnice case study. IEEE Trans Syst Man Cybern Part C Appl Rev 38(4):497–509 (July 2008)
22. Cabrera JCF, Coello Coello CA (2010) Micro-MOPSO: a multi-objective particle swarm optimizer that uses a very small population size. In: Nedjah N, dos Santos Coelho L, de Macedo de Mourelle L (eds) Multi-objective swarm intelligent systems. Theory & experiences. Studies in computational intelligence. Springer, Berlin, pp 83–104 ISBN 978-3-642-05164-7
23. Giannakoglou KC, Kampolis IC (2010) Multilevel optimization algorithms based on metamodel- and fitness inheritance-assisted evolutionary algorithms. In: Tenne Y, Goh C-K (eds) Computational intelligence in expensive optimization problems. Springer, Berlin, pp 61–84. ISBN 978-3-642-10700-9
24. Gonzalez LF, Périaux J, Srinivas K, Whitney EJ (2006) A generic framework for the design optimisation of multidisciplinary uav intelligent systems using evolutionary computing. In: AIAA paper 2006-1475, 44th AIAA aerospace science meeting and exhibit, Reno, Nevada, 9–12 January 2006
25. Simon H, Barone L, While L, Hingston P (2005) A scalable multi-objective test problem toolkit. In: Coello Coello CA, Aguirre AH, Zitzler E (eds) Evolutionary multi-criterion optimization. Third international conference,EMO 2005, Guanajuato, México, Mar 2005, Lecture notes in computer science. Springer, Heidelberg, pp 280–295
26. Huband S, Hingston P, Barone L, While L (2006) A review of multiobjective test problems and a scalable test problem toolkit. IEEE Trans Evol Comput 10(5):477–506
27. Jain AK, Murty MN, Flynn PJ (1999) Data clustering: a review. ACM Comput Surv 31(3):264–323
28. Jin X, Reynolds RG (1999) Using knowledge-based evolutionary computation to solve nonlinear constraint optimization problems: a cultural algorithm approach. In: 1999 congress on evolutionary computation, Washington, DC, July 1999, IEEE Service Center, pp 1672–1678
29. Jin Y (2005) A comprehensive survey of fitness approximation in evolutionary computation. Soft Comput 9(1):3–12
30. Kim Y, Gotoh K, Toyosada M, Park J (2002) Micro-genetic algorithms (μGAs) for hard combinatorial optimisation problems. In: The 12th international offshore and polar engineering conference 2002 (ISOPE 2002), Kitakyushu, Japan, 26–31 May 2002. International society of offshore and polar engineers, pp 230–236
31. Joshua K, David C (2003) Properties of an adaptive archiving algorithm for storing nondominated vectors. IEEE Trans Evol Comput 7(2):100–116

32. Knowles J, Nakayama H (2008) Meta-modeling in multiobjective optimization. In: Branke J, Deb K, Miettinen K, Slowinski R (eds) Multiobjective optimization. Interactive and evolutionary approaches. Lecture notes in computer science. Springer, Berlin, pp 245–284
33. Knowles JD, Corne DW (2000) Approximating the nondominated front using the pareto archived evolution strategy. Evol Comput 8(2):149–172
34. Krishnakumar K (1989) Micro-genetic algorithms for stationary and non-stationary function optimization. SPIE Proc Intell Control Adapt Syst 1196:289–296
35. Becerra RL, Coello Coello CA (2005) Optimization with constraints using a cultured differential evolution approach. In: Beyer H-G et al (eds) Genetic and evolutionary computation conference (GECCO'2005). ACM, Washington, DC, pp 27–34. ISBN 1-59593-010-8
36. Becerra RL, Coello Coello CA (2006) Solving hard multiobjective optimization problems using ε-constraint with cultured differential evolution. In: Runarsson TP, Beyer H-G, Edmund B, Merelo-Guervós JJ, Whitley LD, Yao X (eds) Parallel problem solving from nature—PPSN IX, 9th international conference, Reykjavik, Iceland, September 2006. Lecture notes in computer science. Springer, Heidelberg, pp 543–552
37. Landa-Becerra R, Santana-Quintero LV, Coello Coello CA (2008) Knowledge incorporation in multi-objective evolutionary algorithms. In: Ghosh A, Dehuri S, Ghosh S (eds) Multi-objective evolutionary algorithms for knowledge discovery from data bases. Springer, Berlin, pp 23–46
38. Langer H, Pühlhofer T, Baier H (2004) A multi-objective evolutionary algorithm with integrated response surface functionalities for configuration optimization with discrete variables. In: AIAA paper 2004–4326, 10th AIAA/ISSMO symposium on multidisciplinary analysis and optimization conference, Albany, New York, 30 August–1 September 2004
39. Lara A, Sanchez G, Coello Coello CA, Schütze O (2010) Hcs: a new local search strategy for memetic multi-objective evolutionary algorithms. IEEE Trans Evol Comput 14(1):112–112
40. Lara A, Schütze O, Coello Coello CA (2013) On gradient-based local search to hybridize multi-objective evolutionary algorithms. In: Tantar E, Tantar A-A, Bouvry P, Moral PD, Legrand P, Coello Coello CA, Schütze O (eds) EVOLVE -A bridge between probability, set oriented numerics and evolutionary computation, chapter 9. Studies in computational intelligence. Springer, Heidelberg, pp 305–332. ISBN 978-3-642-32725-4
41. Lee DS, Gonzalez LF, Srinivas K, Periaux J (2007) Multi-objective robust design optimisation using hierarchical asynchronous parallel evolutionary algorithms. In: AIAA paper 2007-1169, 45th AIAA aerospace science meeting and exhibit, Reno, Nevada, 8–11 January 2007
42. Lee DS, Gonzalez LF, Periaux J, Srinivas K (2008) Robust design optimisation using multi-objective evolutionary algorithms. Comput Fluids 37:565–583
43. Lim D, Jim Y, Ong Y-S, Bernhard S (2010) Generalizing surrogate-assisted evolutionary computation. IEEE Trans Evol Comput 14(3):329–355
44. López AL, Coello Coello CA, Schuetze O (2010) A painless gradient-assisted multi-objective memetic mechanism for solving continuous bi-objective optimization problems. In: 2010 IEEE Congress on evolutionary computation (CEC'2010), Barcelona, Spain, 18–23 July 2010, IEEE Press, pp 577–584
45. Loshchilov I, Schoenauer M, Sebag M (2010) Dominance-based pareto-surrogate for multi-objective optimization. In: Deb K, Bhattacharya A, Chakraborti N, Chakroborty P, Das S, Dutta J, Gupta SK, Jain A, Aggarwal V, Branke J, Louis SJ, Tan KC (eds) Simulated evolution and learning, 8th international conference, SEAL 2010, Kanpur, India, 1–4 December 2010. Lecture notes in computer science, vol. 6457. Springer, Heidelberg, pp 230–239
46. Mendoza JE, López ME, Coello Coello CA, López EA (2009) Microgenetic multiobjective reconfiguration algorithm considering power losses and reliability indices for medium voltage distribution network. IET Gener Transm Distrib 3(9):825–840
47. Mendoza J, Morales D, López R, López E, Vannier J-C, Coello Coello CA (2007) Multi-objective location of automatic voltage regulators in a radial distribution network using a micro genetic algorithm. IEEE Trans Power Syst 22(1):404–411
48. Nebro AJ, Luna F, Talbi E-G, Alba E (2005) Parallel multiobjective optimization. In: Alba E (ed) Parallel metaheuristics. Wiley, New Jersey, pp 371–394. ISBN 13-978-0-471-67806-9

49. Pawlak Z (1982) Rough sets. Int J Comput Inform Sci 11(1):341–356
50. Pawlak Z (1991) Rough sets: theoretical aspects of reasoning about data. Kluwer Academic, Dordrecht. ISBN 0-471-87339-X
51. Pilát M, Neruda R (2012) An evolutionary strategy for surrogate-based multiobjective optimization. In: 2012 IEEE congress on evolutionary computation (CEC'2012), Brisbane, Australia, 10–15 June 2012, IEEE Press, pp 866–872
52. Pilato C, Palermo G, Tumeo A, Ferrandi F, Sciuto D, Lanzi PL (2007) Fitness inheritance in evolutionary and multi-objective high-level synthesis. In: 2007 IEEE congress on evolutionary computation (CEC'2007), Singapore, September 2007, IEEE Press, pp 3459–3466
53. Ray T, Isaacs A, Smith W (2009) Surrogate assisted evolutionary algorithm for multi-objective optimization. In: Pandu RG (ed) Multi-objective optimization techniques and applications in chemical engineering. World Scientific, Singapore, pp 131–152
54. Ray T, Smith W (2006) A surrogate assisted parallel multiobjective evolutionary algorithm for robust engineering design. Eng Optim 38(8):997–1011
55. Reyes-Sierra M, Coello Coello CA (2006) Dynamic fitness inheritance proportion for multi-objective particle swarm optimization. In: Keijzer M et al (ed) 2006 Genetic and evolutionary computation conference (GECCO'2006), vol. 1, Seattle, Washington, July 2006. ACM Press, pp 89–90. ISBN 1-59593-186-4
56. Sierra MMR (2006) Use of coevolution and fitness inheritance for multiobjective particle swarm optimization. PhD thesis, Computer science section, Department of Electrical Engineering, CINVESTAV-IPN, Mexico, August 2006
57. Reynolds RG (1994) An introduction to cultural algorithms. In: Sebald AV, Fogel LJ (eds) Proceedings of the third annual conference on evolutionary programming. World Scientific, New Jersey, pp 131–139
58. Reynolds RG, Chung C-J (1997) A cultural algorithm framework to evolve multi-agent cooperation with evolutionary programming. In: Ep '97: Proceedings of the 6th international conference on evolutionary programming VI. Springer, Heidelberg, pp 323–334
59. Reynolds RG, Chung C-J (1997) Knowledge-based self-adaptation in evolutionary programming using cultural algorithms. In: Proceedings of 1997 IEEE international conference on evolutionary computation (ICEC 97), pp 71–76, 1997
60. Reynolds RG, Michalewicz Z, Cavaretta M (1995) Using cultural algorithms for constraint handling in GENOCOP. In: McDonnell JR, Reynolds RG, Fogel DB (eds) Proceedings of the fourth annual conference on evolutionary programming. MIT Press, Cambridge, pp 298–305
61. Ribas PC, Yamamoto L, Polli HL, Arruda LVR, Neves-Jr F (2013) A micro-genetic algorithm for multi-objective scheduling of a real world pipeline network. Eng Appl Artif Intell 26(1):302–313
62. Santana-Quintero LV, Montano AA, Coello Coello CA (2010) A review of techniques for handling expensive functions in evolutionary multi-objective optimization. In: Tenne Y, Goh C-K (eds) Computational intelligence in expensive optimization problems. Springer, Berlin, pp 29–59. ISBN 978-3-642-10700-9
63. Santana-Quintero LV, Coello Coello CA , Hernández-Díaz AG (2008) Hybridizing surrogate techniques, rough sets and evolutionary algorithms to efficiently solve multi-objective optimization problems. In: 2008 Genetic and evolutionary computation conference (GECCO'2008), Atlanta, July 2008. ACM Press, pp 763–764. ISBN 978-1-60558-131-6
64. Santana-Quintero LV, Hernández-Díaz AG, Molina J, Coello Coello CA, Caballero R (2010) DEMORS: a hybrid multi-objective optimization algorithm using differential evolution and rough sets for constrained problems. Comput Oper Res 37(3):470–480
65. Santana-Quintero LV, Ramírez N, Coello Coello C (2006) Multi-objective particle swarm optimizer hybridized with scatter search. In: Gelbukh A, Reyes-Garcia V (eds) MICAI 2006: advances in artificial intelligence, 5th mexican international conference on artificial intelligence, November 2006. Lecture notes in artificial intelligence, vol 4293. Springer, Mexico, pp 294–304
66. Santana-Quintero LV, Ramírez-Santiago N, Coello Coello CA (2008) Towards a more efficient multi-objective particle swarm optimizer. In: Bui LT, Alam S (eds) Multi-objective optimization

in computational intelligence: theory and practice. Information Science Reference, Hershey, pp 76–105. ISBN 978-1-59904-498-9

67. Santana-Quintero LV, Ramírez-Santiago N, Coello Coello CA, Luque JM, Hernández-Díaz AG (2006) A new proposal for multiobjective optimization using particle swarm optimization and rough sets theory. In: Runarsson TP, Beyer H-G, Burke E, Merelo-Guervós JJ, Whitley LD, Yao X (eds) Parallel problem solving from nature-PPSN IX, 9th international conference, Reykjavik, Iceland, September 2006. Lecture notes in computer science, vol 4193. Springer, Heidelberg, pp 483–492
68. Sasaki D, Obayashi S, Nakahashi K (2002) Navier-stokes optimization of supersonic wings with four objectives using evolutionary algorithm. J Aircr 39(4):621–629
69. Sastry K, Goldberg DE, Pelikan M (2001) Don't evaluate, inherit. Proceedings of genetic and evolutionary computation conference. Morgan Kaufmann, Burlington, pp 551–558
70. Sharma D, Collet P (2010) GPGPU-compatible archive based stochastic ranking evolutionary algorithm (G-ASREA) for multi-objective optimization. In: Schaefer R, Cotta C, Kołodziej J, Rudolph G (eds) Parallel problem solving from nature-PPSN XI, 11th International conference, proceedings Part II, September 2010. Lecture Notes in Computer Science, vol 6239. Springer, Poland, pp 111–120
71. Smith RE, Dike BA, Stegmann SA (1995) Fitness inheritance in genetic algorithms. In: SAC '95: proceedings of the 1995 ACM symposium on applied computing, New York, NY, 1995, ACM Press, pp 345–350
72. Szollos A, Smid M, Hajek J (2009) Aerodynamic optimization via multi-objective micro-genetic algorithm with range adaptation, knowledge-based reinitialization, crowding and epsilon-dominance. Adv Eng Softw 40(6):419–430
73. Tagawa K, Shimizu H, Nakamura H (2011) Indicator-based differential evolution using exclusive hypervolume approximation and parallelization for multi-core processors. In: 2011 Genetic and evolutionary computation conference (GECCO'2011), Dublin, Ireland, 12–16 July 2011. ACM Press, pp 657–664
74. Talbi E-G, Cahon S, Melab N (2007) Designing cellular networks using a parallel hybrid metaheuristic on the computational grid. Comput Commun 30(4):498–713
75. Khaled Ahsan Talukder AKM, Kirley M, Buyya R (2009) Multiobjective differential evolution for scheduling workflow applications on global grids. Concurrency Comput-Pract Exp 21(13):1742–1756
76. Tiwari S, Fadel G, Deb K (2011) AMGA2: improving the performance of the archive-based micro-genetic algorithm for multi-objective optimization. Eng Optim 43(4):377–401
77. Tiwari S, Koch P, Fadel G, Deb K (2008) AMGA: An archive-based micro genetic algorithm for multi-objective optimization. In: 2008 Genetic and evolutionary computation conference (GECCO'2008), Atlanta, July 2008. ACM Press, pp 729–736. ISBN 978-1-60558-131-6
78. Pulido GT, Coello Coello CA (2003) The micro genetic algorithm 2: towards online adaptation in evolutionary multiobjective optimization. In: Fonseca CM, Fleming PJ, Zitzler E, Deb K, Thiele L (eds) Evolutionary multi-criterion optimization. Second international conference, EMO 2003, April 2003. Lecture notes in computer science, vol 2632. Springer, Portugal, pp 252–266
79. Pulido GT, Coello Coello CA (2004) Using clustering techniques to improve the performance of a particle swarm optimizer. In: Deb K et al (eds) Genetic and evolutionary computation-GECCO 2004. Proceedings of the genetic and evolutionary computation conference, part I, Seattle, Washington, June 2004. Lecture notes in computer science, vol 3102. Springer, Heidelberg, pp 225–237
80. Van Luong T, Melab N, Talbi E-G (2011) GPU-based approaches for multiobjective local search algorithms. A case study: the flowshop scheduling problem. In: Merz P, Hao J-K (eds) Evolutionary computation in combinatorial optimization, 11th European conference, EvoCOP 2011, Torino, Italy, 27–29 April 2011. Lecture notes in computer science, Vol. 6622. Springer, Heidelberg, pp 155–166
81. Voutchkov I, Keane AJ, Fox R (2006) Robust structural design of a simplified jet engine model, using multiobjective optimization. In: AIAA Paper 2006–7003, Portsmouth, Virginia, 6–8 September 2006

82. Wang Y-N, Wu L-H, Yuan X-F (2010) Multi-objective self-adaptive differential evolution with elitist archive and crowding entropy-based diversity measure. Soft Comput 14(3):193–209
83. Martínez SZ, Coello Coello CA (2008) A proposal to hybridize multi-objective evolutionary algorithms with non-gradient mathematical programming techniques. In: Rudolph G, Jansen T, Lucas S, Poloni C, Beume N (editors) Parallel problem solving from nature-PPSN X, Dortmund, Germany, September 2008. Lecture notes in computer science, Vol. 5199. Springer, Heidelberg, pp 837–846
84. Martínez SZ, Coello Coello CA (2012) A direct local search mechanism for decomposition-based multi-objective evolutionary algorithms. In: 2012 IEEE congress on evolutionary computation (CEC'2012), Brisbane, Australia, 10–15 June 2012. IEEE Press, pp 3431–3438
85. Martínez SZ, Coello Coello CA (2013) A hybridization of MOEA/D with the nonlinear simplex search algorithm. In: Proceedings of the 2013 IEEE symposium on computational intelligence in multicriteria decision making (MCDM'2013), Singapore, 16–19 April 2013. IEEE Press, pp 48–55
86. Martínez SZ, Coello Coello CA (2013) Combining surrogate models and local search for dealing with expensive multi-objective optimization problems. In: 2013 IEEE congress on evolutionary computation (CEC'2013), Cancún, México, 20–23 June 2013. IEEE Press, pp 2572–2579
87. Martínez SZ, Coello Coello CA (2013) MOEA/D assisted by RBF networks for expensive multi-objective optimization problems. In: 2013 Genetic and evolutionary computation conference (GECCO'2013), New York, July 2013. ACM Press, pp 1405–1412. ISBN 978-1-4503-1963-8
88. Zhu W, Yaseen A, Li Y (2011) DEMCMC-GPU: an efficient multi-objective optimization method with gpu acceleration on the fermi architecture. New Generation Comput 29(2):163–184
89. Zitzler E, Deb K, Thiele L (2000) Comparison of multiobjective evolutionary algorithms: empirical results. Evol Comput 8(2):173–195